



Web Services with Spring 5

Introduction. Working with Git

About me



Trayan Iliev

- CEO of IPT – Intellectual Products & Technologies
<http://www.iproduct.org>
- Oracle® certified programmer 15+ Y
- end-to-end reactive fullstack apps with [Java](#), [ES6+](#), [TypeScript](#), [Angular](#), [React](#) and [Vue.js](#)
- 12+ years IT trainer: [Spring](#), [Java EE](#), [Node.js](#), [Express](#), [GraphQL](#), [SOA](#), [REST](#), [DDD](#) & [Reactive Microservices](#)
- Voxxed Days, jPrime, Java2Days, jProfessionals, BGOUG, BGJUG, DEV.BG speaker
- Organizer RoboLearn hackathons and IoT enthusiast

Key Takeaways:

- Achieve a solid level understanding and practical experience using [core Java technologies and APIs](#) – [Java IO/NIO/NIO2](#), [generics](#), [functional programming with java - lambdas and Stream API](#), [Multithreading](#), [JDBC](#);
- Embrace the rich opportunities for [rapid web and REST/Web API application development](#) using [Spring 5](#) and [Spring Boot](#) modules;
- Solid level understanding and professional use of [ORM](#) with [Hibernate/JPA](#);
- Develop, deploy, optimize, secure, and test production grade web applications, services and clients with in [Spring Boot](#), [Hibernate](#), [Spring MVC](#) & [REST](#);
- Hands-on experience with [Spring Boot](#), [Hibernate](#), [JPA](#), [Spring MVC](#), [Spring Data](#), [Spring Security](#), [Spring Boot tests](#), [Spring MVC Testing framework](#), [JUnit](#), [Mockito](#).

Course Topics and Homeworks - I

1. Basic Git workflow – 3 h
2. OOP with Java SE – 8h - [Homework 1: Algorithmic and data structure problem](#); [Homework 2: OOP programming problem](#)
3. Main Java APIs, Design Patterns – 16 h - [Homework 3: Functional programming problem using Stream API](#); [Homework 4: Choosing a REST API course project to implement using DDD and Spring – project specification](#); [Homework 5: Choosing a REST API course project to implement using DDD and Spring – project specification](#); [Homework 6: Extracting JavaDoc comments from project source files](#)
4. Introduction to Spring – 2h - [Homework 7: Configure initial version of your course project using Spring Boot](#)

Course Topics and Homeworks - II

5. Spring beans configuration and Dependency Injection (DI) – 6h - [Homework 8: Implement model classes, in-memory domain repositories and services for the chosen course project](#)
6. Spring AOP and SpEL basics – 3 h
7. Creating REST controllers – 6h - [Homework 9: Implement REST Controllers for the chosen course project](#)
8. Bean Validation API – 1 h
9. Error handling – 2 h - [Homework 10: Implement validation and error handling for the chosen course project](#)
10. Spring Boot & Hexagonal Architecture – 1h

Course Topics and Homeworks - III

- 11. Creating domain services – 3 h - [Homework 11: Implement domain services for the chosen course project](#)
- 12. Creating repositories with Spring Data JPA and Hibernate – 10 h - [Homework 12: Implement DAO \(persistence\) layer for the chosen course project using Spring Data JPA and MySQL](#)
- 13. Using Data Transfer Objects (DTOs) and model mapping – 3h
- 14. Testing Spring Boot applications and components with JUnit 5 – 4h
- 15. Documenting and testing Web APIs using Swagger/OpenAPI
- 16. Spring Security for the REST API with JWT – 4h - [Homework 13: Implement JWT-based security for the chosen course project using Spring Security](#)
- 17. Recap and final projects discussion – 2h

Course Dates:

- 12, 14, 26, 28, юли;
- 1, 3, 5, 8, 10, 12, 15, 17, 19, 22, 24, 26, 29, 31 август
- 2, 5, 7, 9, 12, 14, 16, 19, 21, 23, 26, 28 септември

Time Schedule

- 18:30 – 19:45 – Block 1
- 19:45 – 20:00 – Break
- 20.00 – 21.15 – Block 2

Software to Install

- IntelliJ IDEA - <https://www.jetbrains.com/idea/download/#section=windows>
- MySQL Community Server + MySQL Workbench - <https://dev.mysql.com/downloads/mysql/>
- Chrome Web Browser

Where to Find The Code and Materials?

Java Academy projects and examples are available @GitHub:

<https://github.com/iproduct/java-spring-academy-2022.git>

Agenda for This Session

- Basic version control with Git

Git

Materials from: <https://git-scm.com/book/en/v2>

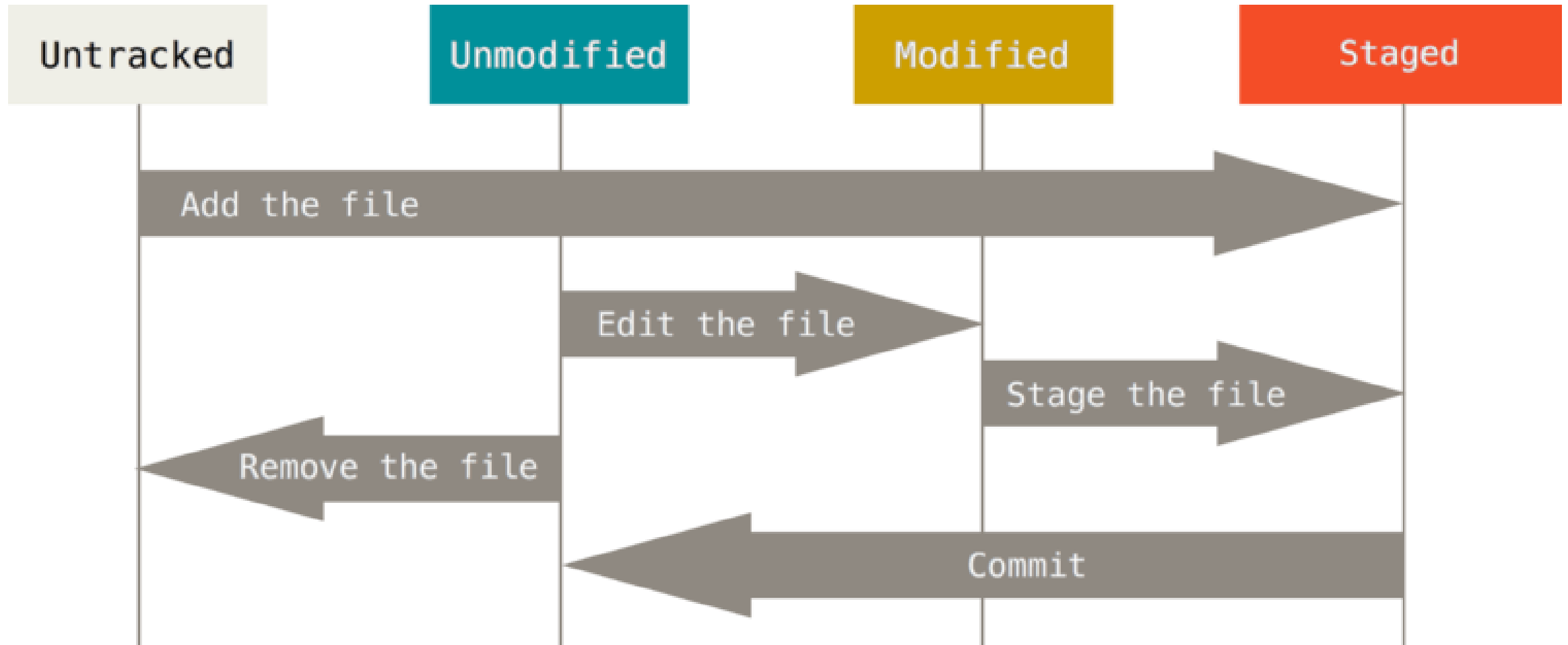
License: [Creative Commons Attribution Non Commercial Share Alike 3.0 license](#)



Social Coding using Git

- Version control systems and collaborative coding: **CVS**, **SVN**, **Git**
- Version control system – allows saving the code changes in a structured and manageable way, with ability to recover previous code state (rollback), experiments (branches), and changes synchronization (merge)
- A distinctive feature of Git is that the changes are kept locally in a form of momentary pictures (snapshots), instead of saving the list of changes – allows fast operations.
- Three stages: **Modified** → **Staged** → **Committed**

Social Coding using Git



Main Git Commands (1)

- Configuring Git

```
$ git config --global user.name "John Smith"
```

```
$ git config --global user.email jsmith@company.com
```

- Help information for a command

```
$ git help <command_verb>
```

- Creating new repository in an existing directory

```
$ git init
```

- Local cloning of a git repository

```
$ git clone <repository_url> [<local_folder>]
```

Main Git Commands (2)

- Adding new files – Staging и Commit

```
$ git add *.java
```

```
$ git add README.txt
```

```
$ git commit -m "initial commit of MyProject"
```

- Information about the status of the files in the project

```
$ git status
```

- Showing changes in the files

```
$ git diff
```

- Ignoring files – file **.gitignore**

```
$ cat .gitignore
```


Main Git Commands (3)

- Removing files

```
$ git rm README.txt
```

```
$ git commit -m "removing README file from project"
```

- Renaming files

```
$ git mv README.txt README
```

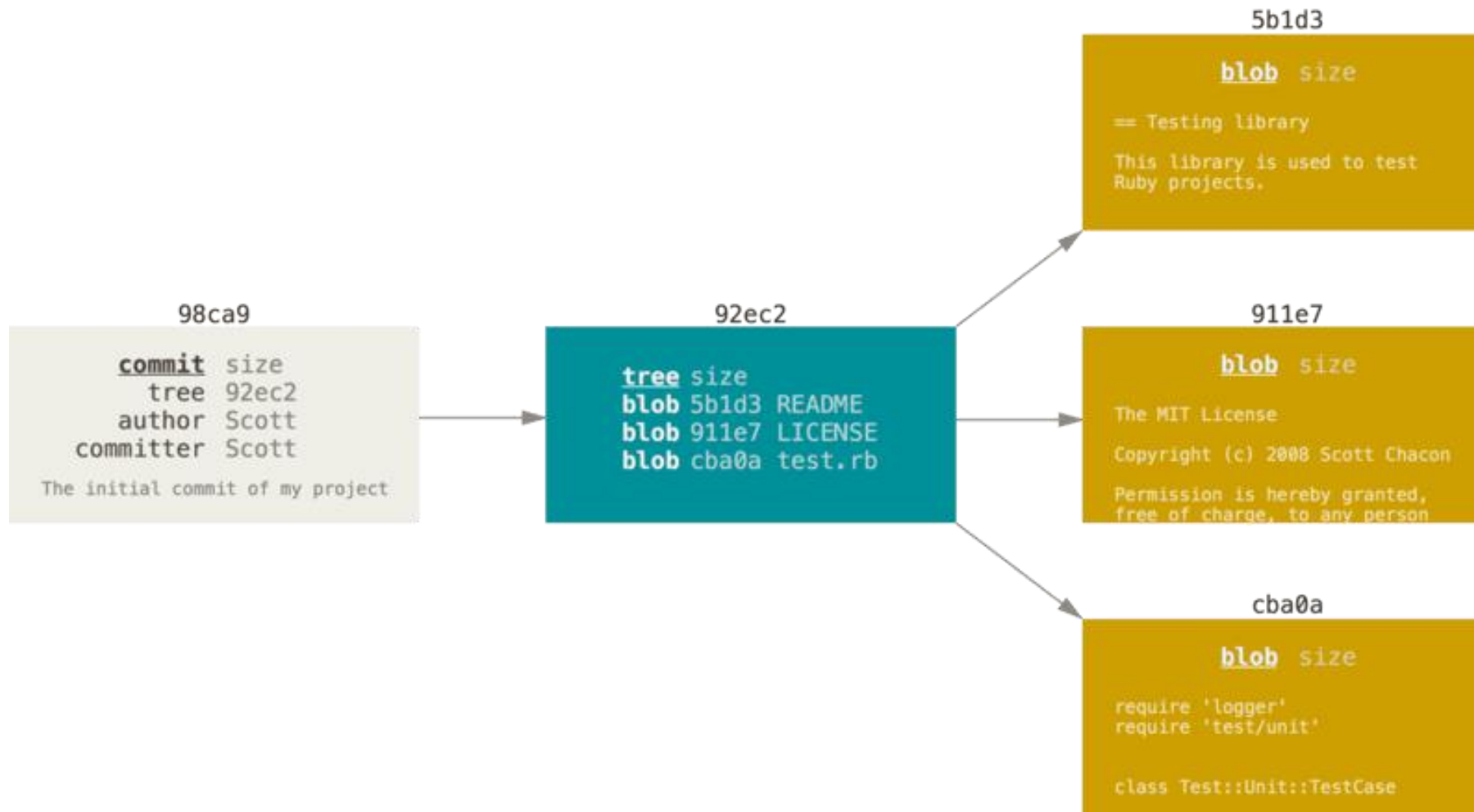
- For more information:

<http://git-scm.com/book/en/Git-Basics-Recording-Changes-to-the-Repository>

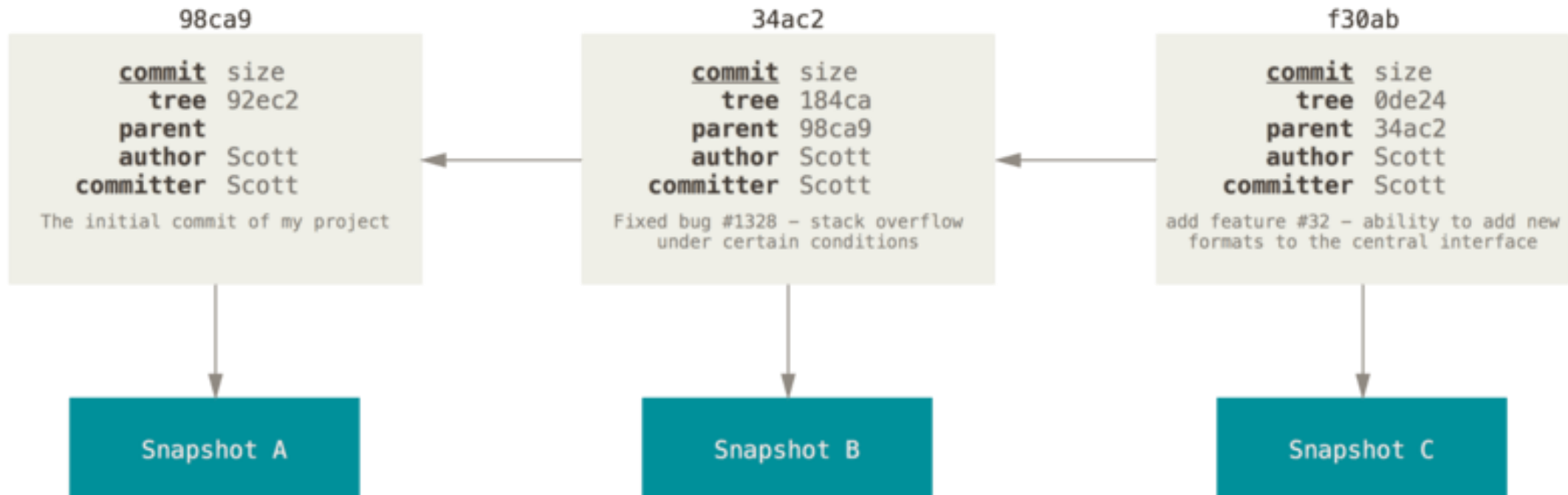
- Example Git project:

<https://github.com/iproduct/java-fundamentals-2022.git>

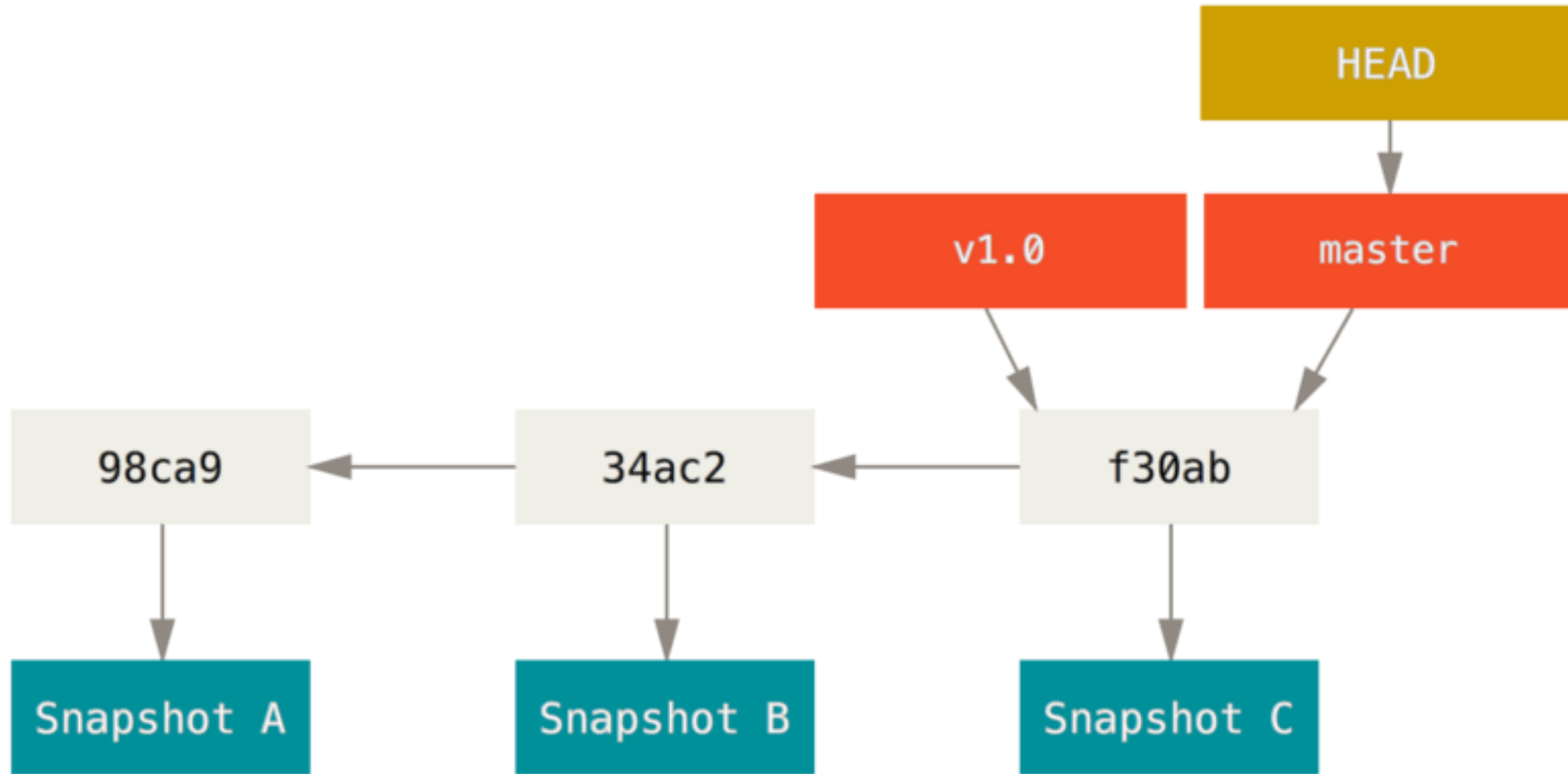
Git Blobs



Git Commits

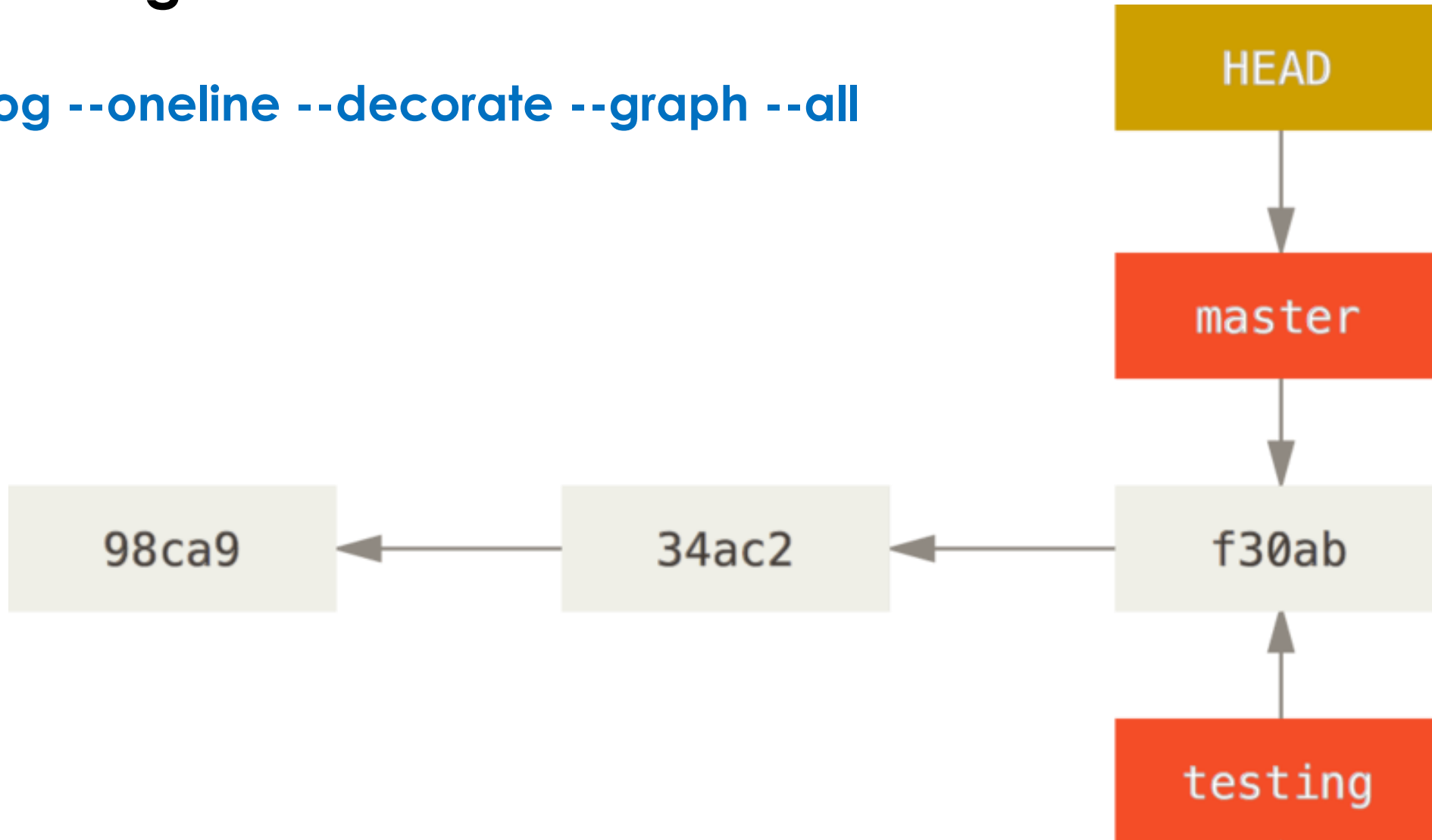


Head and Branches



Branching

`git log --oneline --decorate --graph --all`



Switching Branches -

<https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell>



```
D:\Course_Java_Web_Development\git\course-git-lab>git reset --hard e0ea918
HEAD is now at e0ea918 Merge branch 'test' into main
```

```
D:\Course_Java_Web_Development\git\course-git-lab>git log --oneline --decorate --graph --all
```

```
* e0ea918 (HEAD, origin/main, main) Merge branch 'test' into main
|
| * 74083d8 (origin/test, test) exit command added
| * 32f1102 PrintAllProductsCommand added
|/
|
| * 0dca372 (tag: v1.4) conflict resolved - both products added
|/
| * b4692b6 (tag: v1.1) Update Main.java
| * aecdc9f product 1 changed
|/
|
| * a2295b4 Merge remote-tracking branch 'refs/remotes/origin/main' into main
|/
| * a0b619b Update README.md
| * 3f2f9ad book description changed, .idea folder ignored
|/
|
| * 1cccd12 .gitignore ignores java unit tests
| * e147ef0 .gitignore ignores java unit tests
| * b116047 .gitignore ignores java unit tests
| * d011b18 .gitignore ignores java unit tests
| * 74607c8 .gitignore ignores java unit tests
| * 7b3729c initial project commit
```

```
D:\Course_Java_Web_Development\git\course-git-lab>git checkout 0dca372 .
```

```
D:\Course_Java_Web_Development\git\course-git-lab>git checkout e0ea918 .
```

```
D:\Course_Java_Web_Development\git\course-git-lab>
```

* ef1c500 Initial commit

D:\Course_Java_Web_Development\git\java-spring-academy-2022>gitk --all --date-order

D:\Course_Java_Web_Development\git\java-spring-academy-2022>gitk --all --date-order

The screenshot shows the Git GUI (gitk) interface. The top panel displays a commit history graph on the left and a list of commits on the right. The selected commit is SHA1 ID: 687399c6f81f60cc4035ce93c1ff2103c8bc04f4, which is the initial commit. The bottom panel shows a diff view for the file 01-java-intro/src/model/Person.java. The diff shows the removal of the 'fullName' attribute and the addition of a 'fullName' field in the 'Person' class.

Commit history (SHA1 ID, Date, Time):

- 687399c6f81f60cc4035ce93c1ff2103c8bc04f4 (Initial commit) 2022-07-12 18:16:28
- ca51e9b2522c112391ab25a972348167417b0282 (Experiment using new StringJoiner class) 2022-07-12 21:19:05
- c1b3255069ec4ecb9036a6902408d5326aa4935e (User class added inheriting Person) 2022-07-12 21:28:57
- 88f08341f04c85d903c6ce6c93bfacb394734efd (Person demo added) 2022-07-12 22:27:25
- v0.1.0 (Presentation 2 added) 2022-07-14 17:57:47
- v0.1.1 (Presentation 2 updated) 2022-07-14 18:13:44
- v0.1.0 (Presentation 2 added) 2022-07-14 20:43:39
- v0.1.1 (Presentation 2 updated) 2022-07-14 21:17:43
- v0.1.0 (Presentation 2 added) 2022-07-14 21:26:46
- v0.1.1 (Presentation 2 updated) 2022-07-14 21:28:46

Diff view (01-java-intro/src/model/Person.java):

```
----- 01-java-intro/src/model/Person.java -----
index b1275d3..86b462b 100644
@@ -6,8 +6,6 @@ public class Person {
    private String lastName;
    private int age;

-   private String fullName;
-
    public Person() {
    }
```


Resources

- Pro Git book – <https://git-scm.com/book/en/v2>

Thank's for Your Attention!



Trayan Iliev

IPT – Intellectual Products & Technologies

<http://iproduct.org/>

<https://github.com/iproduct>

<https://twitter.com/trayaniliev>

<https://www.facebook.com/IPT.EACAD>